

MEYERTONS  
HOOD  
KIVLIN  
KOWERT  
& GOETZEL

A PROFESSIONAL CORPORATION

PATENTS, TRADEMARKS, COPYRIGHTS & UNFAIR COMPETITION

700 LAVACA, SUITE 800  
AUSTIN, TEXAS 78701  
TELEPHONE (512) 853-8800  
FACSIMILE (512) 853-8801

**FAX**

To: Examiner Hichman Foud

From: Paul Seegers

Fax: 571-270-2463

Pages: 17 (including cover)

Phone: 571-270-1463

Date: July 27, 2009

Re: U.S. Application No. 10/660,188

Phone: (512) 853-8878

Please see the attached proposed response to the Office Action of 4/14/09.

THIS FACSIMILE TRANSMITTAL AND THE DOCUMENTS ACCOMPANYING THIS FACSIMILE TRANSMITTAL CONTAIN CONFIDENTIAL INFORMATION INTENDED ONLY FOR THE USE OF THE INDIVIDUAL NAMED ABOVE. IF YOU ARE NOT THE INTENDED RECIPIENT YOU ARE NOTIFIED THAT THIS COMMUNICATION MAY BE SUBJECT TO THE ATTORNEY-CLIENT OR WORK-PRODUCT PRIVILEGE AND THAT THE DISSEMINATION, DISTRIBUTION OR COPYING OF THIS COMMUNICATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS COMMUNICATION IN ERROR, PLEASE IMMEDIATELY NOTIFY US BY TELEPHONE (COLLECT) TO ARRANGE FOR RETURN OF THE DOCUMENTS. RECEIPT BY ANYONE OTHER THAN THE INTENDED RECIPIENT IS NOT A WAIVER OF ANY ATTORNEY-CLIENT OR WORK-PRODUCT PRIVILEGE.

**PATENT****IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Inventor:	David E. Mayhew	§	Atty.Dkt.No.:	6257-14502
		§		
Serial Number:	10/660,188	§	Examiner:	Foud, Hicham B.
		§		
Filing Date:	September 11, 2003	§	Group/Art Unit:	2419
		§		
Title:	ADVANCED SWITCHING ARCHITECTURE	§	Conf. No.	5820
		§		
		§		
		§		
		§		

**RESPONSE TO OFFICE ACTION MAILED APRIL 14, 2009**

This paper is submitted in response to an Office Action of April 14, 2009, to further highlight why the application is in condition for allowance.

Please amend the case as listed below.

**IN THE SPECIFICATION:**

Please amend the paragraph [0024] to recite as follows:

When an origin constructs a path, it must supply two values: the turn pool and the bit count (also referred to herein as "turn count"). When routing packets to endpoints the bit count is always initialized to be zero. When routing packets to switches, the bit count must be biased. For a packet to be accepted by a switch its turn count must be 23 when it arrives at the switch. To ensure this necessary condition, an endpoint that wishes to communicate with a switch must set the initial bit count of switch based packets to be the 23 plus the bit size of the active turn pool partition.

**IN THE CLAIMS:**

The following is a current listing of claims and will replace all prior versions and listings of claims in the application. Please amend the claims as follows:

1. (Currently Amended) An apparatus, comprising:

a switch having a plurality of ports, wherein said switch is configured to receive a packet on a first of said plurality of ports, said packet including header data including a first turn value ~~specifying a second of the plurality of ports relative to the first port;~~

wherein said switch is configured, based on an identifier for the first port, the first turn value, said header data and the number of said plurality of ports, to transmit said packet on a the second of said plurality of ports.

2. (Previously Presented) A system, comprising:

a switch having a plurality of ports including a first port and a second port, wherein said switch is configured to receive a packet on said first port, wherein said packet includes header data, said header data comprising a turn pool, wherein said turn pool comprises a plurality of turn values, including a turn value specifying the second port relative to the first port;

wherein the switch is configured, based on said header data and the number of said plurality of ports, to transmit said packet on said second port.

3. (Canceled)

4. (Previously Presented) The system of claim 2, wherein said header data is comprised of a credit length, a bit count, an operation, a Path Identifier (PID) index, a Maximum Transmission Unit (MTU) and an Extended Unique Identifier (EUI).

- 5-12. (Canceled)

13. (Previously Presented) The system of claim 2, wherein said header data further comprises a bit count.

14. (Previously Presented): A switch, comprising:  
a plurality of ports;  
means for receiving a packet on a first of said plurality of ports, wherein said packet includes header data including a plurality of turn values;  
means for using one of said plurality of turn values to determine a second of said plurality of ports on which to transmit said received packet; and  
means for transmitting said packet on said second port.
15. (Currently Amended) A switch, comprising:  
a plurality of ports;  
first means for receiving a packet on a first port of said plurality of ports, said packet comprising packet header data, wherein said packet header data comprises a turn pool, wherein said turn pool comprises a plurality of turn values, one of which specifies a second port of said plurality of ports relative to said first port;  
second means for using said turn pool, a bit count, and the number of said plurality of ports to select said second port on which to transmit said packet; and  
third means for transmitting said packet on said second port.
16. (Canceled) ~~The switch of claim 15, wherein said packet header data further comprises a bit count and said second means is configured to further utilize[[s]] said bit count to select said second port.~~
17. (Currently Amended) The switch of claim 15, further comprising:  
fourth means for modifying said packet header data prior to transmitting said packet.

18. (Previously Presented) A method, comprising:  
receiving, at a switch within a network, an encapsulated packet, wherein said encapsulated packet includes header data that includes a plurality of turn values, and wherein said encapsulated packet is received at first of a plurality of ports of said switch;  
determining a second port of said plurality of ports using said header data and the number of said plurality of ports; and  
transmitting said encapsulated packet from said switch via said second port.
19. (Previously Presented) The method of claim 18, further comprising modifying said header data prior to transmitting via said second port.
20. (Previously Presented) A method of routing a packet from a source to a destination within a fabric having at least one switch, said method comprising:  
receiving an encapsulated packet at a first of a plurality of ports of said at least one switch, wherein the encapsulated packet includes a header including a first turn value that specifies a second of said plurality of ports relative to the first port;  
determining said second port using said header of the encapsulated packet and the number of said plurality of ports; and  
transmitting said encapsulated packet from said at least one switch via said second port.
21. (Currently Amended) The method of claim 20, wherein said ~~packet field data~~ header further comprises a bit count.
22. (Previously Presented) The method of claim 20, further comprising modifying said header prior to transmitting via said second port.
23. (Previously Presented) The method of claim 22, wherein said header further comprises a bit count.
24. (Previously Presented): The method of claim 20, wherein said fabric comprises a plurality of switches, and said method further comprises repeating said receiving,

determining and transmitting at various ones of the plurality of switches with corresponding ones of a plurality of turn values associated with the packet until said packet reaches said destination.

25. (Currently Amended) The method of claim 21, said header further comprising a turn pool including a plurality of turn values that includes said first turn value, wherein said destination is configured to use said turn pool and bit count of said packet ~~are usable by said destination~~ to create a second header to encapsulate a second packet to be routed from said destination to said source.

26. (Currently Amended) The method of claim 23, said header further comprising a turn pool including a plurality of turn values that includes said first turn value, wherein said destination is configured to use said turn pool and bit count of said packet ~~are usable by said destination~~ to create a second header to encapsulate a second packet to be routed from said destination to said source.

27. (Previously Presented) The apparatus of claim 1, said header data including a plurality of turn values that includes said first turn value, wherein each of the plurality of turn values corresponds to a respective network device within a path for said packet and specifies an output port of its respective network device relative to an input port of the respective network device, and wherein a given one of the respective network devices in the path is configured to transmit said packet on an output port of the given device that is specified by the corresponding one of the plurality of turn values.

28. (Previously Presented) The method of claim 20, wherein said header includes a turn pool including a plurality of turn values that includes said first turn value.

**REMARKS:**

Claims 1, 2, 4 and 13-28 were pending in the application. Claims 1, 15, 17, 21, 25, and 26 have been amended. Claim 16 has been canceled. Therefore, claims 1, 2, 4, 13-15, and 17-28 are now pending in this application.

**Specification Objections**

Paragraph [0014] is objected to because “it is not known what the applicant means by the terms ‘turn credit’ and ‘traffic class credit’ because they are no[t] fields in the header with the above terms.” Office Action at 2. Applicant submits that embodiments of Applicant’s disclosure may be applicable to the PCI express protocol standard. *See* Specification at ¶ [0011]. Applicant further submits that this standard is known to use a “credit-based flow control” in order to ensure that packets are transmitted only when it is known that a buffer is available to receive these packets at the other end. The specification states that the terms “turn credit” and “traffic class credit” do not refer to specific header field names, but rather the “credit type” field of a unicast packet. *See* Specification 4-5 Table 1. Accordingly, “next turn credit” and “traffic class credit,” in some embodiments, refer to different types of credit that may be used for flow control within a switching architecture. “Next turn credit” can broadly refer to credit associated with a “next turn” in a path specification, while “traffic class credit” can refer to credit associated with a particular “traffic class” or priority.

Paragraph [0024] is objected to because “it is not known what the applicant means by the term ‘turn count’.” *Id.* As can be seen from the context of paragraph [0024], the terms “bit count” and “turn count” are used interchangeably. Applicant has amended paragraph [0024] accordingly to clarify this usage.

**Claim Objections**

Claim 13 is objected to as being a duplicate claim of claim 4. Applicant respectfully disagrees and submits that the claims recite different elements capturing different ranges of scope. For example, claim 4 recites that the “header data is comprised of a credit length, a bit count, an operation, a Path Identifier (PID) index, a Maximum Transmission Unit (MTU) and an Extended Unique Identifier (EUI)” (emphasis added). In contrast, claim 13 recites that the



“header data further comprises a bit count” and does not require the other types of information recited in claim 4. As but one non-limiting example, claim 13 might be applicable to embodiments that use header data including a “bit count” but not “an Extended Unique Identifier (EUI)” as recited in claim 4. Accordingly, claim 4 and 13 are clearly not duplicate claims.

Claim 21 is objected to because it lacks antecedent basis for “said packet field data.” Applicant has amended claim 21 to recite “said header,” which has antecedent basis in claim 20.

Claim 25 and 26 is objected to for reciting “usable.” Applicant has amended these claims to remove this term.

### Double Patenting

With respect to double patenting issues raised by the Examiner, see Office Action at 3, Applicant respectfully requests that this rejection be held in abeyance until the claims in the identified co-pending application are found to be otherwise in condition for allowance.

### Section 112 Rejections

#### *Written Description*

Claims 1, 2, 4, and 13-28 are rejected under 35 U.S.C. 112, first paragraph, for reciting a “turn value specifying a second port,” when the specification indicates that “the output port is specified by [a] turn value, input port and the number N and not only the turn value as claimed.” See Office Action at 5. While Applicant disagrees that the identified claims include “new matter,” Applicant has amended claim 1 to recite that “said switch is configured, based on an identifier for the first port, the first turn value, and the number of said plurality of ports, to transmit said packet on a second of said plurality of ports.” Claims 2, 14, 15, 18, and 20 have been amended in a similar manner. Applicant respectfully request removal of these rejections.

#### *Omitting Essential Elements*

Claims 1, 2, 4 and 13-28 are rejected under 35 U.S.C. 112, second paragraph, as “being incomplete for omitting essential elements, such omission amounting to a gap between the elements.” Office Action at 5. In particular, the Examiner asserts that the identified claims omit the following (allegedly) essential features: 1) “the claimed switch is required to support path routing and only to forward the packet according to the path that is contained in the packet

header,” 2) “the transmission of the path-routed packets depend also on the bit count” and 3) “the output port is specified as ‘An output port number = ([input\_port\_number+turn\_value+1] modulo [N.sup2+1]).’” Office Action 5-6. Applicant respectfully disagrees with these rejections.

Applicant submits that in order for matter to be considered essential, it must be “disclosed to be essential to the invention as described in the specification or in other statements of record.” MPEP 2172.01. Applicant submits that the specification does not indicate that the features identified by the Examiner meet this standard.

As to the first feature listed above, the Examiner cites paragraph [0018] of Applicant’s specification, which recites:

All ExAS nodes are required to support path routing. A path specifies the position of the terminus relative to the origin, and is assigned to the ExAS header by the origin of the packet. Nodes are required only to forward the packet according to the path that is contained in the ExAS packet header.

While this passage states that “*ExAs nodes* are required” to support various features, the specification is not limited to the use of “ExAs” (which is tied to the PCI Express standard). See Specification ¶ [0011] (referring to “the PCI Express Advanced Switching (PCI ‘ExAS’) architecture” and stating that embodiments of the disclosure “provide[] for an extensible switching fabric framework for encapsulation of virtually any protocol,” including “the PCI Express”). In other words, embodiments of the disclosure may be applicable to protocols other than the PCI Express protocol. As such, Applicant submits that, even though the specification uses the term “required” when describing “ExAS nodes,” the specification does not teach or suggest that the features recited in paragraph [0018] need be present in every possible embodiment.

As to the Examiner’s suggestion that the functionality of a “bit count” is omitted from the claims, Applicant notes that nothing in the specification indicates that this particular feature is “essential.” Applicant does note that various ones of the pending claims do recite a “bit count”—e.g., claim 15 recites “using said turn pool, a bit count, and the number of said plurality of ports to select said second port on which to transmit said packet” (emphasis added).

Finally, as to the specific formula recited by the Examiner for specifying the output port, Applicant submits that the specification does not identify this formula as being “essential.” In any event, Applicant has amended claim 1 to recite “said switch is configured, based on an

identifier for the first port, the first turn value, and the number of said plurality of ports, to transmit said packet on a second of said plurality of ports.”

*Indefiniteness*

Claims 1, 2, 4, and 13-28 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite. In particular, the Examiner rejected claims 1, 2, 14, 15, 18, and 20 for reciting “that the turn value specifies the second port then transmitting the packet based on [the] turn value/header and number of ports.” As noted above, claim 1 now recites that “said switch is configured, based on an identifier for the first port, the first turn value, and the number of said plurality of ports, to transmit said packet on a second of said plurality of ports.” Claims 2, 14, 15, 18, and 20 have been amended in a similar manner. Such amendments are believed to address the Examiner’s concerns. Applicant has also canceled claim 16, rendering any rejection of this claim moot.

**CONCLUSION:**

Applicant respectfully submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicant hereby petitions for such extension.

The Commissioner is authorized to charge any fees that may be required, or credit any overpayment, to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account No. 501505/6257-14502/DMM.

Respectfully submitted,

Date: \_\_\_\_\_

By: \_\_\_\_\_  
Dean M. Munyon  
Reg. No. 42,914

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P. O. Box 398  
Austin, Texas 78767  
(512) 853-8847

# FPGA and Programmable Logic JOURNAL

www.fpgajournal.com

TRENDS  
TECHNOLOGY  
TOOLS  
TECHNIQUES

## PCI Express Design Considerations Platform ASIC vs. FPGA Design Efficiency

### PCI Express Design Considerations

Platform ASIC vs. FPGA Design Efficiency

by Greg Martin, RapidChip Technical Marketing, LSI Logic

Implementing a high-speed PCI-Express core is a complex task, even for the most seasoned engineers. To further complicate matters, the choice of implementation technology can play a significant role in the final design characteristics. When evaluating FPGA and Platform ASIC technologies, there are a number of key considerations.

### Smaller Footprint

A typical 8-lane (32Gbps aggregate) PCI Express interface can be implemented with a 64-bit data path running at 250MHz or with a 128-bit data path running at 125MHz. It is extremely difficult to successfully implement any reasonably complicated digital design (with ~20 logic levels) at 150MHz in an FPGA. Reaching anywhere near 250MHz for such designs is not possible, even with the latest 90nm FPGAs. Therefore an x8 PCI Express core implemented in an FPGA will require a 128-bit datapath clocked at 125MHz. By contrast, when implemented in

a Platform ASIC, the same core can easily achieve 250MHz allowing the smaller, more efficient 64-bit datapath implementation to be used.

In a Platform ASIC implementation, all of the data paths will be half the width of an FPGA implementation. Since these data paths comprise a large portion of the entire design it will have a major impact on overall gate count. A typical FPGA implementation uses approximately 60% more logic resources than a Platform ASIC implementation.

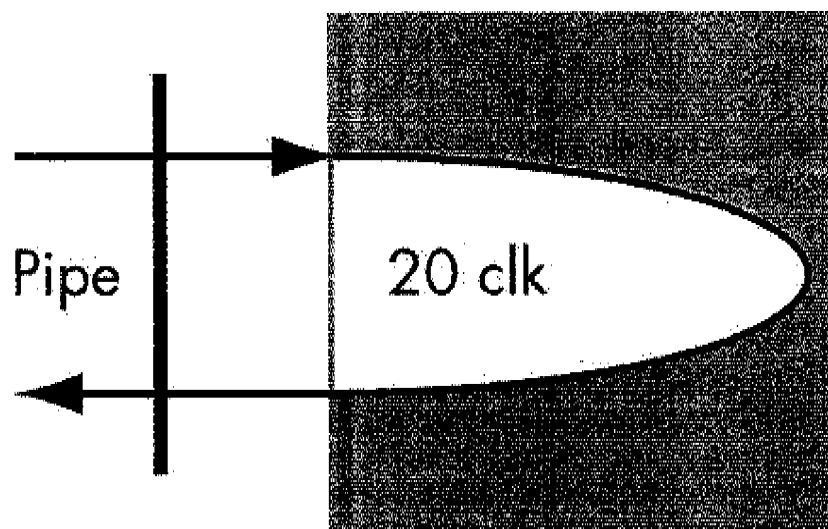
Additionally, buffer sizes in the FPGA implementation are often larger than a Platform ASIC implementation. Not only are the buffers wider, in many cases they must be deeper to cope with latency effects.

### Reduced Latency

The latency of a controller greatly influences the overall performance of the

PCI Express interface and thus the entire system. The round trip latency of a design is a very important metric. It is measured from the PIPE Rx to the PIPE Tx, going across the physical, link and transaction layers. A typical PCI Express controller configuration will have ~15- 25 clock cycle round trip latency.

Consider the case of a controller with 20 clock cycles round trip latency. When implemented at 125MHz in an FPGA, the 20-clock cycle latency is  $20 \times 8\text{ns} = 160\text{ns}$ . The same core implemented at 250MHz in a Platform ASIC has only  $20 \times 4\text{ns} = 80\text{ns}$  clock cycle latency. The 100% additional latency suffered by an FPGA implementation is a major reason for the superior performance of Platform ASICs.



### Better Link Utilization

The reduced latency of Platform ASICs vs. FPGAs can also translate into superior link utilization. For example, consider the utilization of a PCI Express egress link with a standard-cell ASIC link-partner, in an Intel north bridge system. Figure 2a shows the PCI Express transmit path implemented in a Platform ASIC. Figure 2b shows the PCI Express transmit path implemented in an FPGA.

In the transmit datapath, the PCI Express core sends packets to the standard link partner buffer. When packets leave this buffer, credits are released back to the PCI Express core.

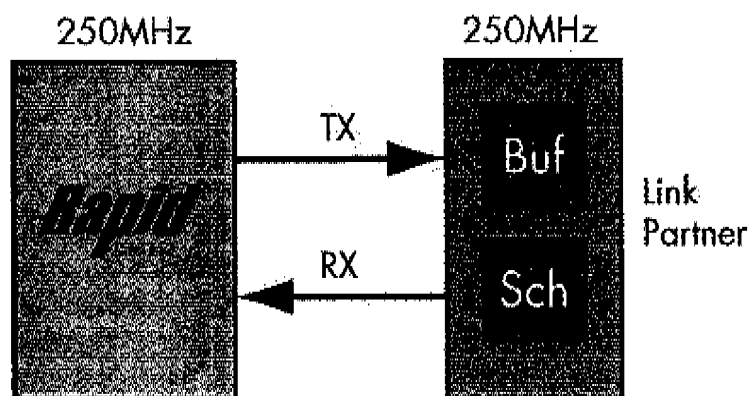


Figure 2a

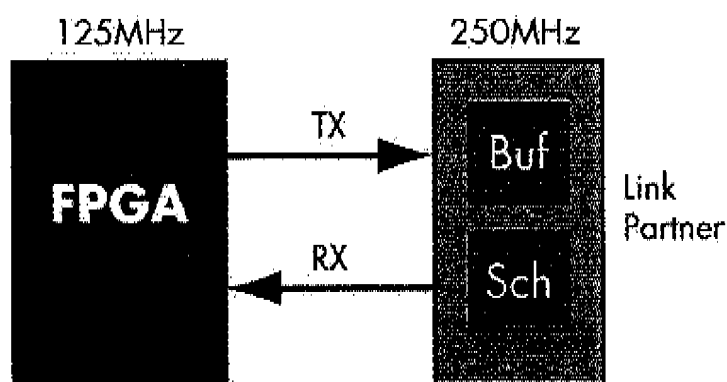


Figure 2b

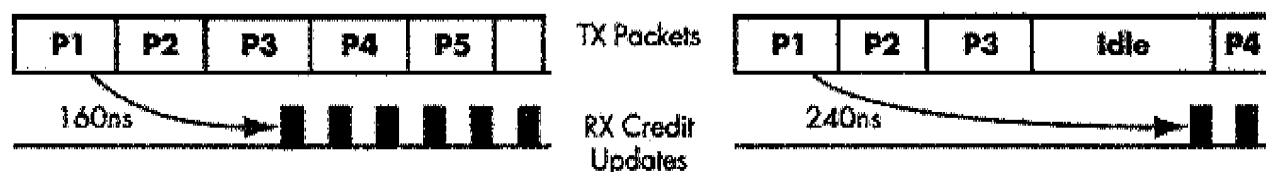
The size of the receive buffer in the link partner and the latency in receiving the credit back to the PCI Express core determines how efficiently the link is utilized.

The fixed size of the Virtual Channel (VC) buffer on the receiving standard-cell ASIC link partner will typically be optimized with the expectation of connection to a similar standard-cell ASIC like device. Thus it will work most efficiently when connected to something with corresponding latency similar to that of a standard-cell ASIC.

If the end-to-end latency involved in sending a packet from the PCI Express core and receiving the credit back is much more than the typical number assumed in the above buffer size estimation then the link will start idling due

to credit starvation. This starvation occurs when the receiving buffer is not large enough to absorb the additional end-to-end latency.

A simple comparison between the Platform ASIC and FPGA implementations is shown in Figures 2c and 2d. This analysis is simplified by excluding the effects of packet size, credit release policy etc. Figure 2c shows how the Platform ASIC implementation continuously sends packets. Its ASIC-like latency allows credits to be received back fast enough to avoid starvation. In contrast Figure 2d shows how an FPGA has to wait much longer for credit updates to occur causing the link to go idle.



This example only considers the case of posted-write packet types, although the effects also apply to other packet types and multi-VC cases. A major component of the credit latency path is the controller's internal delay. Lets assume the round trip latency inside the link-partner is 20 cycles (at 250MHz). The most significant portion of the end-to-end credit return delay is the sum of the round trip latencies of the both controllers. I.e.  $20 \times 4 \text{ ns}$  for the link-partner plus  $20 \times 4 \text{ ns}$  for the Platform ASIC implemented PCI Express core. This gives a total of 160ns.

The same setup for an FPGA implementation of the PCI Express core will take  $20 \times 4 \text{ ns}$  for the link-partner plus  $20 \times 8 \text{ ns}$  for the FPGA, giving a total of 240ns. If the buffer in the link-partner has been designed to cover only the first case latency of 160ns, then the link utilization for the FPGA implementation will be 33% lower.

## Reduced Buffer Size

The receive path to a PCI Express core also has similar link utilization considerations.

In an FPGA implementation, the receive VC buffer size must be increased by 50% to absorb the increase in end-to-end latency (240ns instead of 160ns, for the above example). This means the Platform ASIC implementation



requires a reduced buffer size compared with an FPGA implementation. If the FPGA receive buffer size is not increased, the receive path into the PCI Express core will also suffer from utilization problems.

### Increased Overall Performance

In addition to the local credit starvation and link utilization issues, the increased latency of an FPGA implementation affects other areas of system performance. Figures 3a and 3b highlight how latency affects the read performance in a system. For a given number of outstanding reads from a node, any increased latency in receiving a response adds significant waiting time for the read initiator. This reduces overall read bandwidth. If the read data contains assembly code to be executed or data-packets to be processed, then the efficiency of such processes will also be significantly reduced.



### Conclusion

For complex, high-speed applications such as PCI Express, even the fastest 90nm FPGAs lack sufficient performance. The workarounds to compensate for this lower performance have wide reaching implications on the final system behavior. Both latency and link utilization are degraded in the slower FPGA-based implementation. In many cases this will ultimately slow down the entire system. Additional resources are also required for an FPGA implementation.

The ASIC-like characteristics of a Platform ASIC give it a clear performance advantage over FPGAs. The Latency and Link-utilization of a PCI Express core are similar to that of a standard cell ASIC and therefore optimal for this application. Having these characteristics is especially important when the link-partner is designed for connection to a similarly low-latency partner.

*by Greg Martin, RapidChip Technical Marketing, LSI Logic*

*August 23, 2005*